1   1. A method for distributing data between a source and destination in a network, the
2   method comprising the steps of:
3   defining, n number of paths between the source and the destination;
4   splitting the data into n number of blocks
5   independently transmitting the n blocks from the source to the destination; and
6   reassembling, at the destination, the n blocks into the data.

1   2. The method according to claim 1 wherein the independent transmission of the n
2   blocks is not started simultaneously.

1   3. The method according to claim 1 wherein the size of the n blocks remains constant
2   for the duration of the transmission.

1   4. The method according to claim 1 wherein the size of the n blocks varies during the
2   duration of the transmission.

1   5. The method according to claim 1 wherein the n number of paths is determined using
2   multicast, depth-first-search technique.

1   6. The method according to claim 1 wherein the n number of paths is determined using
2   multicast spanning tree technique.

1   7. The method according to claim 1 wherein the n number of paths is determined using
2   multicast arborescence technique.

1   8. The method according to claim 1 further comprising the step of:
2   receiving, at the source, a request for a data transfer to the destination.

1   9. A method of file transfer in a computer-based communication network by utilizing an
2   overlay network composed of cooperating servers on computer hosts, said computer
3   hosts connected to said communication network, wherein each said server contains
4   instructions which, when executed by said server, cause said server to process and
5   forward data via the transport layer to other servers on said overlay network without
6   modifying the native data transport protocol at transport or lower layers, said method
7   comprising the steps of:
8   defining a first data forwarding path between two servers, said path comprising of
9   concatenation of overlay links, each of said links established via transport
10   layer between two said servers in said overlay network;
11   defining a second data forwarding path between two servers, said path different
12   from the first data forwarding path;
13   dividing the data file in at least two sub-files, first sub-file and second sub-file;
14   and
15   sending first and second sub-files over the first and second data forwarding
16   paths, respectively.

1   10. The method according to claim 9 further comprising the steps of:

2   replicating each sub-file in one or more intermediate servers on said data

3   forwarding paths and forwarding the sub-file to the next server in the path;

4   and

5   assembling the first and the second sub-files at the second server.

1   11. The method of claim 9, wherein said sending first and second sub-files over the first

2   and second data forwarding paths are not started simultaneously.

1   12. The method of claim 10, wherein said sub-files are transmitted from a single source

2   node to a plurality of destination nodes, and said paths form a data forwarding tree,

3   wherein the intermediate nodes of the tree copy the sub-files reaching them from the

4   incoming overlay link to the plurality of outgoing overlay links.

1   13. The method of claim 9, wherein said sub-files are transmitted from a plurality of

2   source nodes to a plurality of destination nodes, and said forwarding paths form data

3   forwarding trees, wherein the intermediate nodes of the tree copy the data reaching

4   them from the incoming link to the plurality of outgoing links.

1   14. The method of claim 9, wherein the data forwarding and processing resources are

2   reserved, said resources including one or more metrics from the following group:

3   bandwidth of overlay link, processing load of the server.

1   15. The method of claim 9, wherein the data forwarding and processing resources are

2   determined from network monitoring, said resources including one or more metrics

3   from the following group: bandwidth of overlay link, processing load of the server.

1   16. The method of claim 9, wherein the said overlay network has static topology and

2   resources, said resources including one or more metrics from the following group:

3   bandwidth, processing load.

1   17. The method of claim 9, wherein the said overlay network has dynamic topology and
2   resources, said resources including one or more metrics from the following group:
3   bandwidth, processing load.

1   18. The method of claim 9, wherein the data forwarding paths are computed in a single
2   server.

1   19. The method of claim 9, wherein the data forwarding paths are computed in several
2   servers, with subsequent coordination of computed results.

1   20. The method of claim 9, wherein the divisions of the data file into sub-files remain
2   constant for the duration of the file transfer.

1   21. The method of claim 9, wherein the divisions of the data file into sub-files change
2   during the file transfer.

1   22. The method of claim 11, wherein the data forwarding tree is constructed using
2   multicast depth-first-search method.

1   23. The method of claim 11, wherein the data forwarding tree is constructed using
2   multicast spanning tree method.

1   24. The method of claim 11, wherein the data forwarding tree is constructed using
2   multicast arborescence method.

1   25. The method of claim 9, wherein the established communications protocols include
2   one or more protocols selected from the following group: Internet Protocol, http, ftp
3   SSL, TCP reliable UDP using erasure coding.

1   26. The method of assembly of claim 12, wherein the assembly is done at overlay
2   network layer, said method comprising the steps of:
3   encoding a monotonically increased sequence number in each packet header at
4   data source.
5   queuing data packet in a sink buffer in destination for each data transport
6   session.
7   scanning a sink buffer and selecting a data packet that matches the current
8   receiving sequence number maintained at the destination.
9   clocking out available data packets in the sink buffer selected by step c);
10   increasing the receiving sequence number at the destination.
11   delivering the available data packet to the application client; and
12   repeating the method for all sink buffers.

1   27. The method of assembly of claim 12, wherein the assembly is done at the overlay
2   application layer by the application, said method comprising the steps of:
3   encoding an application-specific data object identifier into a data frame;

4   extracting the application-specific data object identifier at the destination; and

5   reassembling the application data object according to the data object identifier.

1   28. The method of data transfer of claim 9, said method comprising the steps of:

2   using an explicit or implicit method to setup the forwarding table in each overlay
3   node;
4   forwarding the data by looking up the forwarding table installed by step a) at each
5   node;
6   scheduling data transport at the data source and at each intermediate node for
7   each path or tree according to their QoS specifications; and
8   slowing down a particular data flow on a path or tree by using backpressure.

1   29. The method of claim 9, wherein the transport of data partitions to a plurality of
2   destination nodes is a combined coordination of transport initiated by the sender
3   (push) and transport initiated by the receivers (pull).

1   30. An overlay network apparatus for determining optimized paths and trees for
2   transmitting data from a source to destinations within a computer-based
3   communications network, the communications network being characterized by one
4   or more established communications protocols, the apparatus comprising:
5   a set of one or more intermediate nodes, the intermediate nodes being operable
6   to transmit and receive data in conformance with the established
7   communications protocols;
8   path and trees discovery means, responsive to a request for transmitting data
9   from the sources to the destinations, operable to discover network resources
10   on virtual links between the sources and the destinations passing through one
11   or more of the intermediate nodes, the paths being derived by means of one
12   or more existing routing mechanisms of the communications network, wherein
13   the intermediate nodes define a virtual topology on top of the computer-based
14   communications network;
15   data processing means for data partitioning, storage and replication at the nodes;
16   and
17   forwarding means for forwarding the data from the sources to the destinations by
18   way of paths and trees, without requiring a modification of the established
19   communications protocols.